OPEN ACCESS

International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-1, November 2019

Highly Parallel Map Reduce Process and Efficient Job Scheduling Methodologies of Big Data Systems.

Suja Cherukullapurath Mana, T. Sasipraba

Abstract: ***This paper studies about various job scheduling methodologies used in big data systems. Map reduce is a highly efficient distributed job processing strategy for big data systems. Job scheduling is a critical task of any big data system as the volume of jobs need to be processed is tremendous. This study will go over the map reduce process in detail. It also reviews various job scheduling methodologies and tries to perform an efficient comparison among these methodologies.***

Keywords: Big data, job scheduling, resource allocation.

I. INTRODUCTION

The term big data refers to the huge volume of data generated day by day. The main characteristics of theses big data systems are its volume, velocity variety and veracity. Volume indicate the huge amount of data being generating, velocity is the speed at which data is being generated. Veracity indicates various issues related to the quality of data .In addition to these four fundamental characteristics, more and more characteristics are being added. Value, validity, volatility, visualization are some example of these characteristics. The following table shows some of the characteristics of big data system and their explanations

Characteristics	Explanation
Volume	Volume indicates the huge amount of data being stored in big data systems
Velocity	Velocity indicates the speed at which data being added to the system
Variety	Big data include both structured and unstructured data
Veracity	The term veracity indicates various data quality issues

Map Reduce is the prominent programming model used in big data systems. It consists of two modules namely map and reduce. Map process will produces key value pairs from data. This key value pairs will be given as an input to the reduce process. The reduce process will combine this key value pairs and finally produce the output. Maps reduce model helps to process huge amount of data effectively. Fig [1] shows the map reduce process

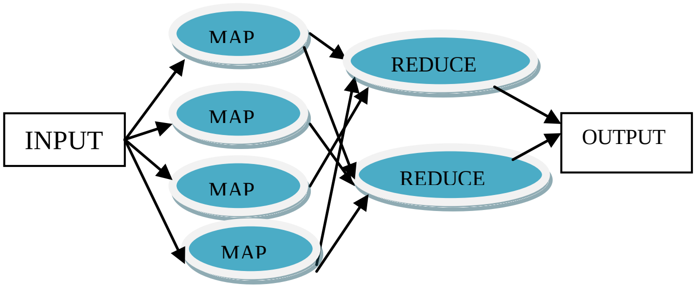

FIG .1. MAP REDUCE PROCESS

Map Reduce is highly efficient in processing huge volume of data. Map process will accept the input data and break the data into set of key value pairs. The reduce task will accept this key value pairs and then combine it to produce the output as shown in Fig [1]. The map reduce process is further divided into steps like splitting, mapping, shuffling as shown in fig .2. Splitting will split the data into small blocks. Mapping process will map the data into different key value pairs. Shuffling will shuffle and arrange these key value pairs. Finally the reducing process will combine these data and produces the final output. The entire process is shown in the diagram below Fig .2.

The diagram in Fig .2 describes the map reduce process in detail. Initially the splitting process will split the input data into different chunks as shown in fig [2]. Then the mapping process will map data into key value pairs. Shuffling process will arrange these key value pairs and provide inputs to reducing process. Reducer will combine the data and produce the output

Revised Manuscript Received on November 30, 2019.
* Correspondence Author

Suja Cherukullapurath Mana*, School of Computing, Sathyabama Institute of Science and Technology,Chennai, India. Email: cmsuja@gmail.com

T. Sasipraba, School of Computing, Sathyabama Institute of Science and Technology,Chennai, India. Email: provc@sathyabama.ac.in

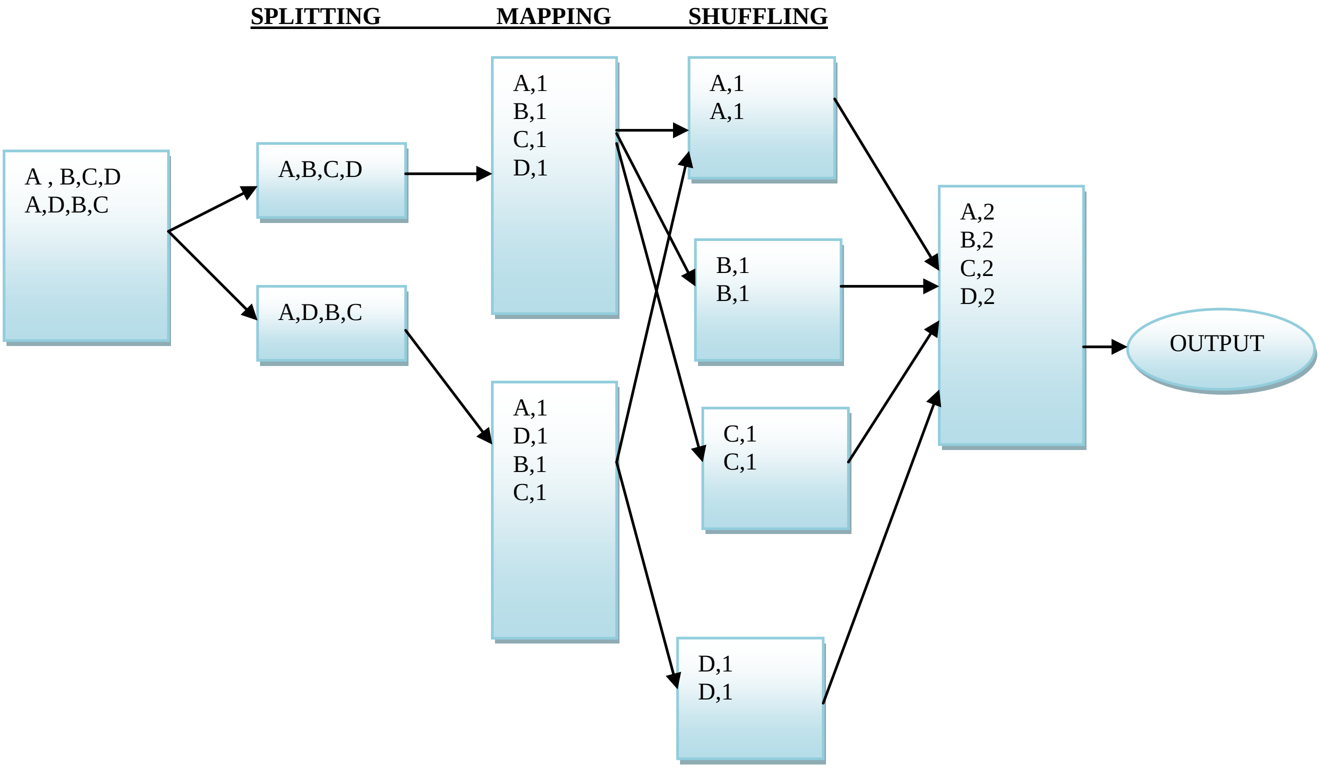

Fig.2 . Detailed Map Reduce process

Hadoop Distributed File System (HDFS) is the file system used by Hadoop. HDFS consist of name nodes and data nodes to do processing [8].It has master slave architecture. HDFS supports the fast processing of data among various Hadoop nodes. The main task of name node is to manage the name space of file system. Each of the file names will be mapped to set of data node block. Cluster configuration management also performed by name node. Name node also stores the metadata and transaction logs. Data node stores the actual

data in local file system. It will send details about existing nodes to name node periodically. Below given diagram Fig .3 shows the architecture of name node and data nodes of a HDFS

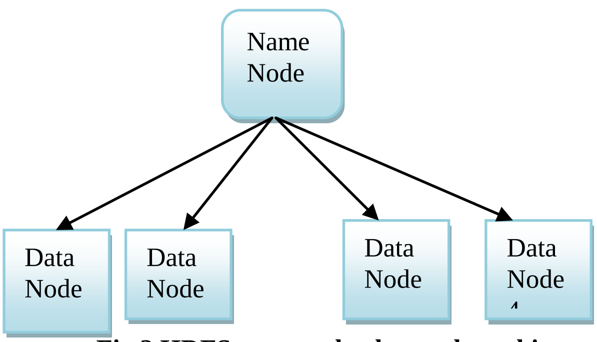

Fig.3 HDFS namenode ,datanode architecture

II. MAP REDUCE SCHEDULING

Job scheduling is an important task in any big data system. It is a critical task which will influence the overall performance of a big data system. It is critical task that helps to reduce the execution time and cost, thereby influences the overall efficiency of system [10]. There are various job schedulers available. This paper will go through some of the job scheduler available as well as some of the scheduling algorithms. Various factors like job cluster heterogeneity, scalability, amount of jobs to be processed will influence the performance of various schedulers [1]. Other factors being considered in this study are resource allocation, optimization methodology, mechanism used to implement the scheduling, resource sharing etc [8]. First in First out (FIFO), Fair scheduler and longest approximation time to end (LATE) is some of the prominent job scheduling policies used in big data system [1].

There are some built in resource management frameworks being used in Hadoop system [8]. Yet Another Resource Negotiator [Yarn], Mesos, Corona is some among them [2]. Yarn is a recourse allocation framework works well in distributed environments. The main concept behind YARN is to separate resource management task from data processing which allows the big data system to concentrate well on data processing [11]. The two important components of Yarn are resource manager and application master. The main task of resource manager is to manage all resources available and the primary task of application master is to work with node manager in monitoring and executing Resources [8]. Fig [4] illustrates main components of Yarn

International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-1, November 2019

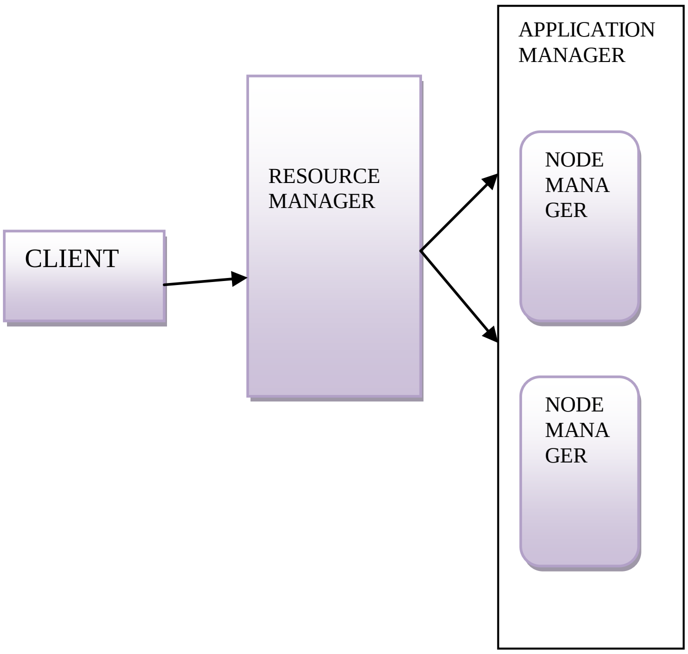

Fig .4 .Main components of Yarn

Mesos is another recourse manager framework developed for efficient allocation of resources in a heterogeneous environment [12]. It is open source software first developed at University of California, Berkeley [8]. Mesos track resources efficiently and intimate the scheduler on availability of resources [13]. It is designed as a two level scheduler with allow inclusion of various pluggable scheduling algorithms [13] . This two level scheduling architecture allows the framework to have the flexibility of selecting scheduling algorithms based on requirements. Fig. 4 shows the architecture of MesoS. The resource manager keeps tracks of the available resources and sends that list of available resource to the big data system. Big data system depending on the need to launch processes accepts these resource offers [8].

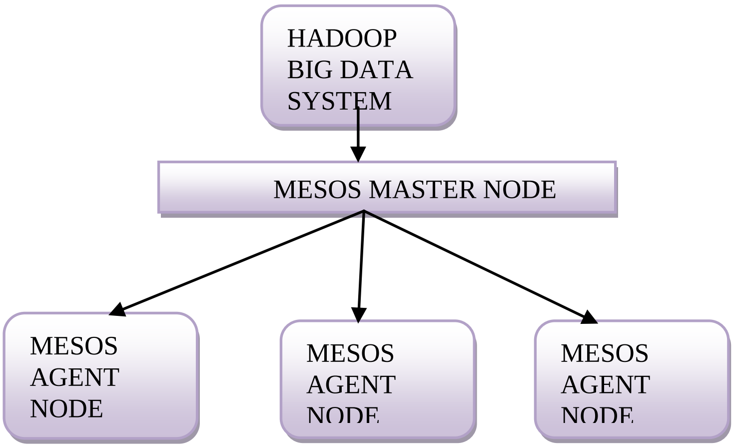

Fig.4 Mesos Architecture

The below given table provides a comparison of Mesos and Yarn schedulers .

Mesos	Yarn
Push based memory/CPU scheduling scheduling	Pull based memory
Mesos need separate security security mechanism	Yarn inherits Hadoop mechanism
Mesos cannot perform application specific scheduling	Yarn is application specific scheduler
Cannot support cluster application	It can support cluster applications
Mesos is highly scalble	Yarn is less scalable

III. SCHEDULING ALGORITHMS

This section will go over some of the map reduce scheduling algorithms like first in first out, delay scheduler, fair scheduler etc

A. First In First Out (FIFO) Scheduler

FIFO is a simple job scheduler. The process which comes first will get the opportunity to process first [1]. It allocates jobs base on the arrival time. FIFO scheduler's performance is better in low load environment. FIFO scheduler is easy implement but some jobs need to starve a large amount of time if the process which currently running is taking a long time to complete [1]. For example a small job the wait time may exceed the total completion time due to the first in first out policy. A FIFO queue is the main component of the scheduler [3]. Jobs are divided into small tasks and are inserted into the FIFO queue [3]. The oldest task (based on arrival time) will be taken from the job queue and will be processed. Job tracker and task tracker nodes are responsible for these operations [3].

B. Fair Scheduler

Fair scheduler ensures the fairness property in scheduling. It ensures that no process will starve for an unlimited period of time. Each job will be ensured to get a minimum amount of processing slots [3]. In the fair scheduler no task will starve for an unlimited amount of time [4]. Jobs will be kept in a job pool and a fair share of processing time will be allocated to each scheduler. Fairness in allocation is ensured in this job scheduling methodology.

C. Longest Approximate Time To End(LATE) Scheduler

LATE scheduler follows a static job allocation policy [4]. This scheduler will identifies a slow running task and assign another task as a backup task for it [5]. Three important policies guide the execution of the LATE scheduler [3].

It will avoid thrashing of tasks, always selects faster nodes for execution, tasks will be assigned priorities to speculate [3]. If the cost to launch the background task is less LATE is considered to an efficient scheduling algorithm [6]. It works on both homogeneous and heterogeneous environments. It is also a preemptive scheduling policy [6]. It ensures fairness in recourse allocation as well.

D. Capacity Scheduler

Capacity scheduler ensures efficient processing if number of tasks is large [7]. This scheduler utilizes queue data structure to implement the scheduling. Each queue will be having designated capacity .If the capacity of a specific queue is not enough it seeks unused recourses from other queues [3]. To efficiently implement the job scheduling re allocation of resources among queues will be performed [3]. Once the job is completed borrowed resources will be given back to the original queue. It is a resource aware static scheduling policy [8]. It works well in homogeneous environments.

IV. CONCLUSION

This paper will perform a brief study on the characteristics of big data and also go over some of the resource management frameworks used in big data systems. This study also analyses some of the scheduling algorithms used in the big data systems. The below given table summarizes some of the characteristics of these algorithms The selection of scheduling algorithm is based on the nature of the job to be performed and also based on the environment of processing. By selecting an efficient job scheduling algorithm maximum efficiency can be achieved.

Name	Policy	Enviornment	Static/Dynamic
FIFO	Scheduling based on arrival time	Homogeneous	Static
Fair Scheduler	Small tasks will run in parallel	Homogeneous	Static
LATE	Tasks will be assigned priorities	Heterogeneous	Static
Capacity Scheduler	Uses designated queues of diffrent capacity	Homogeneous	Static

The selection of scheduling algorithm is based on the nature of the job to be performed and also based on the environment of processing. By selecting an efficient job scheduling algorithm maximum efficiency can be achieved.

REFERENCES

1. Honeem, Mohammad & Kulkarni, Lalit. (2017). An Adaptive MapReduce Scheduler for Scalable Heterogeneous Systems. 10.1007/978-981-10-1678-3_57.
2. Jia-Chun Lin and Ming-Chang Lee, “Performance evaluation of job schedulers on Hadoop YARN,” Concurrency and Computation: Practice and Experience (CCPE), vol. 28, no. 9, 2016, pp. 2711–2728
3. Anjana Sharma, International Journal of Computer Science and Mobile Computing, Vol.4 Issue.12, December- 2015, pg. 171-176
4. S. C. Mana, "A Feature Based Comparison Study of Big Data Scheduling Algorithms," *2018 International Conference on Computer, Communication, and Signal Processing (ICCCSP)*, Chennai,2018,pp.1-3. doi: 10.1109/ICCCSP.2018.8452837
5. Arpitha HV and Shoney Sebastian “Comparative study of Job Schedulers in Hadoop Environment” , International Journal of Advanced Research in Computer Science ,Volume 8 , No-3 March-April 2017
6. M Zaharia et al., “Improving MapReduce Performance in Heterogeneous Environments”, Operating systems design and implementation, Pp 29-42, 2008
7. S Thakur et al., “Dynamic Capacity Scheduling in Hadoop”, International Journal of Computer Applications, vol. 125, Issue 15, 2015.
8. Ibrahim Abaker Targio Hashem, et al.” MapReduce scheduling algorithms: a review” , The Journal of Supercomputing ,December 2018
9. Zaharia M et al (2010) Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling. In: Proceedings of the 5th European Conference on Computer Systems. ACM
10. Chang H et al (2011) Scheduling in MapReduce-like systems for fast completion time. In: 2011Proceedings IEEE INFOCOM. IEEE
11. Vavilapalli VK et al (2013) Apache Hadoop YARN: yet another resource negotiator. In: Proceedingsof the 4th Annual Symposium on Cloud Computing. ACM
12. Hindman B et al (2011) Mesos: a platform for fine-grained resource sharing in the data center. In:NSDI
13. Scott J (2015) A tale of two clusters: Mesos and YARN. [cited 2016 1/6/2016]. http://radar.oreilly.com/2015/02/a-tale-of-two-clusters-mesos-and-yarn.html
14. Jithina Jose, Suja Cherukullapurath Mana, B Keerthi Samhitha , “An Efficient System to Predict and Analyze Stock Data using Hadoop Techniques” , International Journal of Recent Technology and Engineering (IJRTE), Volume-8 Issue-2, July 2019

AUTHORS PROFILE

Ms.Suja Cherukullapurath Mana. Ms.Suja C.M is actively doing research in data science, big data, and decision guidance systems.She is an Assistant Professor at School of Computing, Sathyabama Institute of Science and Technology

Dr.T.Sasipraba :Dr.Sasipraba is working as pro vice chancellor at Sathyabama Institute of Science and Technology